



ARL-TR-7961 • FEB 2017



Installing and Executing Information Object Analysis, Intent, Dissemination, and Enhancement (IOAIDE) and Its Dependencies

by Laurel C Sadler and Christian D Schlesiger

Approved for public release; distribution is unlimited.

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.



Installing and Executing Information Object Analysis, Intent, Dissemination, and Enhancement (IOAIDE) and Its Dependencies

by Laurel C Sadler and Christian D Schlesiger
Computational and Information Sciences Directorate, ARL

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
<p>Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.</p> <p>PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.</p>					
1. REPORT DATE (DD-MM-YYYY) February 2017		2. REPORT TYPE Technical Report		3. DATES COVERED (From - To)	
4. TITLE AND SUBTITLE Installing and Executing Information Object Analysis, Intent, Dissemination, and Enhancement (IOAIDE) and Its Dependencies				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Laurel C Sadler and Christian D Schlesiger				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) US Army Research Laboratory ATTN: RDRL-CII-B 2800 Powder Mill Road Adelphi, MD 20783-1138				8. PERFORMING ORGANIZATION REPORT NUMBER ARL-TR-7961	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT Information Object Analysis, Intent, Dissemination, and Enhancement (IOAIDE) is a novel information framework developed at the US Army Research Laboratory that supports the needs of command and control analysts as well as researchers in both algorithm and tool prototyping. It supports dynamic plugin of analysis modules, for either research or analysis tasks. The framework integrates multiple image processing algorithms, image and video feeds, and a dissemination service for the dissemination and retrieval of information objects, as well as a central database for storage and query of information objects. With the variety of services and development tools required to develop and run the IOAIDE tool, it is rather complicated to deploy the application or set up its development environment. This report is targeted at developers and users with the basic knowledge and understanding of Windows-based computer operations. The report lists the required commercial software packages and describes in intricate detail the steps involved in preparing the IOAIDE application computer for development, execution, and database setup.					
15. SUBJECT TERMS software, installation, applications, integration, instructions					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT UU	18. NUMBER OF PAGES 38	19a. NAME OF RESPONSIBLE PERSON Laurel C Sadler
a. REPORT Unclassified	b. ABSTRACT Unclassified	c. THIS PAGE Unclassified			19b. TELEPHONE NUMBER (Include area code) 301-394-1221

Contents

List of Figures	iv
1. Introduction	1
2. Computer and Software Development Tools Requirements	2
3. Installing the Software for IOAIDE	2
3.1 Load ARL Software	2
3.2 Load ARL Applications	4
3.3 Load the DSPro Software	7
3.4 Update Java Setting in Regedit	22
3.5 Build the IOAIDE Software Packages	24
3.6 Build DSProTest	25
3.7 Copy IOAIDE and Dependencies Shortcuts to Executables	26
4. Execute IOAIDE and Dependencies in Development/Debug Mode	26
5. Execute IOAIDE and Dependencies from Shortcuts and Executables	27
6. Conclusion	30
7. References	31
Distribution List	32

List of Figures

Fig. 1	SVN Checkout...	3
Fig. 2	SVN Checkout textbox	4
Fig. 3	Install ARLApps	5
Fig. 4	ARLApps Setup Wizard	5
Fig. 5	User Account Control (new app)	6
Fig. 6	Unzip files	7
Fig. 7	Move Nomads to C:	8
Fig. 8	System popup window	9
Fig. 9	Visual Studio DSPro project	10
Fig. 10	Connect to Server window	12
Fig. 11	SQL Server Script window	12
Fig. 12	Find and Replace window	13
Fig. 13	MATLAB Compiler Runtime Installer	14
Fig. 14	Visual Studio: ARLImageProcessing.Host	15
Fig. 15	Developer Command Prompt for VS2015	16
Fig. 16	Services	16
Fig. 17	ARL Image Processing Web Server Administration	17
Fig. 18	Microsoft ASP.NET MVC 4 installation	18
Fig. 19	Visual Studio: WeatherDigest.sln	19
Fig. 20	Developer Command Prompt for VS2015	20
Fig. 21	Services	21
Fig. 22	Internet Explorer Weather Digest test	21
Fig. 23	Internet Explorer Weather Digest Twitter test	22
Fig. 24	User Account Control for regedit	22
Fig. 25	Registry Editor	23
Fig. 26	Edit DWORD (32-bit) Value	23
Fig. 27	Visual Studio: ARLImageProcessing.SurfaceApp	25

1. Introduction

Information Object Analysis, Intent, Dissemination, and Enhancement (IOAIDE) is a novel information framework developed at the US Army Research Laboratory (ARL) that supports the needs of command and control analysts as well as researchers in both algorithm and tool prototyping. It supports dynamic plugin of analysis modules, for either research or analysis tasks. The framework integrates multiple image processing algorithms, image and video feeds, and a dissemination service for the dissemination and retrieval of information objects, as well as a central database for storage and query of information objects. IOAIDE provides an environment to run experiments with analysts, human-in-the-loop, to evaluate the ability to enhance the capability of the analyst, reduce the workload of the analyst, and provide visual organization of data based on the value of information (VoI) of the data for the analyst.¹ Future research using the IOAIDE environment will include experiments to evaluate the effectiveness of allowing the analyst to modify the VoI of an information object and then disseminate as a new message with higher VoI or re-disseminate the same message with increased VoI.

Several of the image processing algorithms are made into client applications that can be accessed from an image processing web service² developed following Representational State Transfer (REST) standards by a mobile app, laptop PC, and other devices. Similarly, weather tweets can be accessed via the Weather Digest Web Service, which collects and extracts essential weather information from the National Weather Service and sends the digested information to Twitter. These data can be viewed by the analyst or used by other algorithms that may have been integrated into IOAIDE. Other more central processing unit (CPU)-intensive and time-constrained algorithms are installed directly on the laptop PC or Microsoft Surface Pro.

IOAIDE supports both dissemination and retrieval of information objects through 2 third-party software packages: *Dissemination Service (DisService)*³ and *Proactive Dissemination Service (DSPro)*⁴. DisService is a peer-to-peer, publish-subscribe, store-carry-and-forward middleware designed to support efficient information dissemination and group communication in resource-constrained networks. DSPro extends the functionality of DisService by supporting proactive transmission of information, which realizes mission-context, VoI-based filtering, hence, limiting the amount of data the analyst receives, which is the first step in reducing analyst workload.

With the variety of services and development tools required to develop and run the IOAIDE tool, it is rather complicated to deploy the application or set up its

development environment. This report is targeted at developers and users with the basic knowledge and understanding of Windows-based computer operations. The report lists the required commercial software packages and describes in intricate detail the steps involved in preparing the IOAIDE application computer for development, execution, and database setup.

2. Computer and Software Development Tools Requirements

The user interface and system control software web services were developed in .NET Framework 4.5.2 with Windows Presentation Foundation using Microsoft Visual Studio 2015 (version 14.0). The dissemination service was developed in C++ using Microsoft Visual Studio 2015. The database was designed and is maintained with Microsoft SQL Server Management Server. Basic requirements for the IOAIDE computing environment prior to executing the instructions in this report include a Microsoft Windows device running Windows 10 with Visual Studio 2015 and Microsoft SQL Server 2016 installed.

It is also assumed that the system upon which IOAIDE is to be installed is connected to the Research Development and Engineering Network so as to have access to various resources stored in network storage. One of these resources is an Apache Subversion (SVN) software repository from which most of the software is checked out. Therefore, a SVN client is required: it is assumed that TortoiseSVN has been installed on the target computer system. Certain resources have been archived with the 7-Zip utility application, which are then required to access these archives on the target computer.

3. Installing the Software for IOAIDE

This section describes each step necessary to add or install software and copy necessary video data to the computer to build a basic demonstration/development environment for IOAIDE:

3.1 Load ARL Software

The following are the steps for loading the ARL software:

- 1) Check out the Components and commercial off-the-shelf (COTS) directories from <https://batinfo.ciib-idvrn.arl:8443/svn/ARLProducts/trunk/Components> and <https://batiopnfo.ciib-idvrn.arl:8443/svn/cots/trunk/COTS> to C:\ARLDeveloper\Components and C:\ARLDeveloper\COTS, respectively. If setting up the computer for Open Standard for Unattended Sensors plugins, check out <https://batinfo.ciib-idvrn.arl:8443/svn/ARLProducts/trunk/Java> to

C:\ARLDeveloper\Java, as well. It may be necessary to use the direct IP address 172.18.128.22 in place of batinfo.ciib-idvrn.arl in the previous addresses to reach the repository.

2) To check out from SVN, do the following:

- a) Right click the mouse on the desktop and select SVN Checkout... (Fig. 1). The checkout menu's popup window appears on the screen.

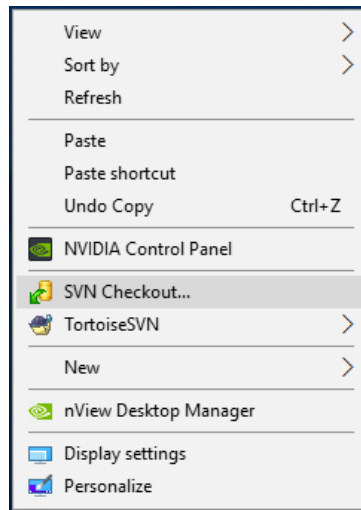


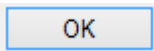
Fig. 1 SVN Checkout...

- b) In the first textbox (Fig. 2), type

https://batinfo.ciib-idvrn.arl:8443/svn/ARLProducts/trunk/Components

- c) In the second textbox (Fig. 2), type

C:\ARLDeveloper\Components

- d) Click  (Fig. 2).

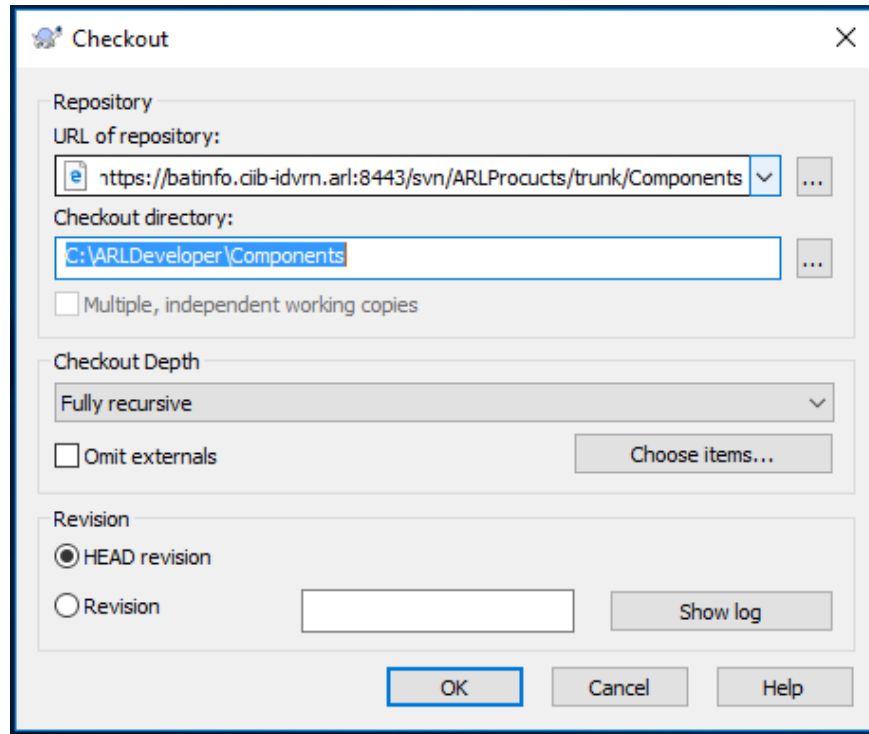



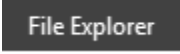


Fig. 2 SVN Checkout textbox

- e) Repeat these steps to check out the additional items listed previously.

3.2 Load ARL Applications

The following are the steps for loading the ARL applications:

- 1) If the Start network  and Stop network  icons are not visible on the desktop, it is necessary to install the ARLApps application. To install ARLApps, perform the following steps:
 - a) Right click the Windows icon  and left click . The File Explorer popup window appears on the screen (Fig. 3). In the textbox at the top of the File Explorer, type (including all spaces)
\\Synologynas\Root\Branch Software\ARL Software\ARLApps
 - b) Double click ARLApps_20161115_1742.msi (Fig. 3).

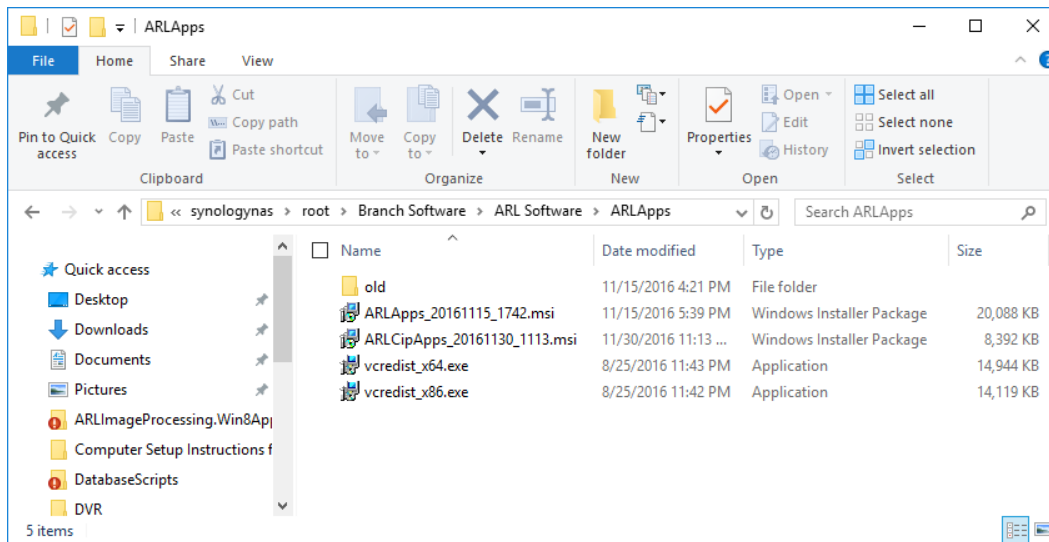


Fig. 3 Install ARLApps

c) The ARLApps Setup Wizard appears on the screen (Fig. 4).

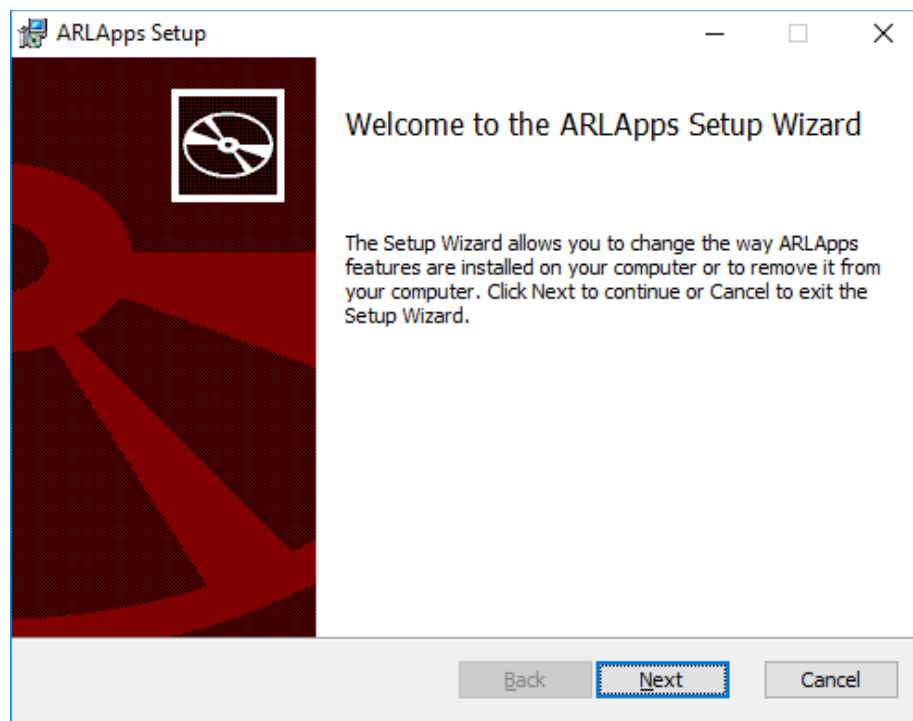
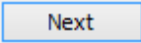



Fig. 4 ARLApps Setup Wizard

d) Click **Next**.

e) Check the ☒ I accept the terms in the License Agreement check box and click **Next**.

- f) The Custom Setup window appears. Click .
- g) The ready to install ARLApps window appears. Click .
- h) The User Account Control Window appears on the screen (Fig. 5).

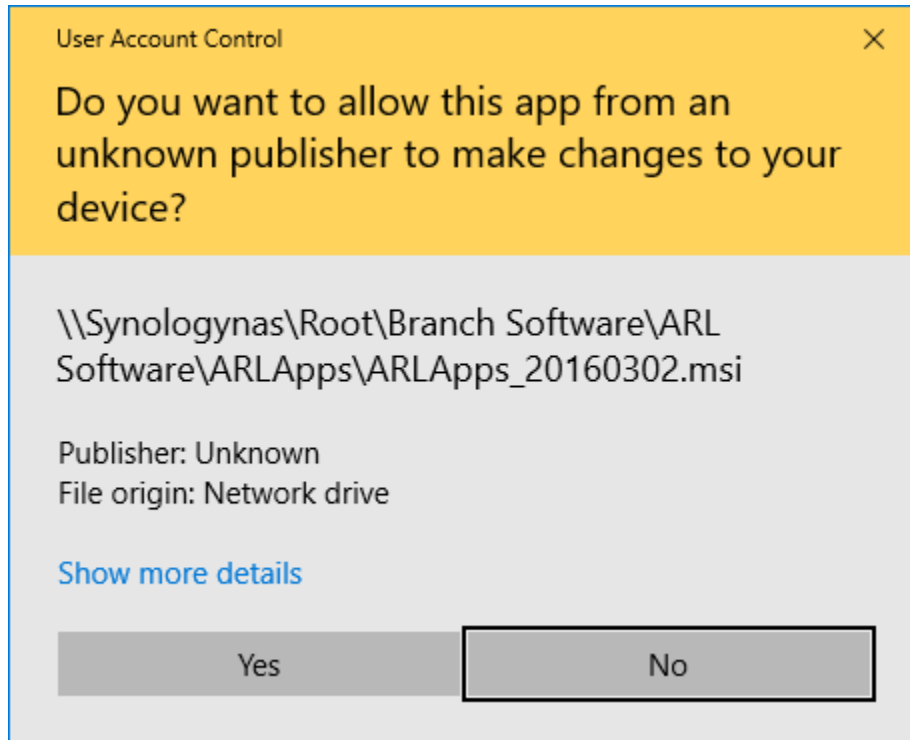
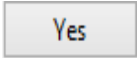
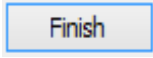




Fig. 5 User Account Control (new app)

- i) Click .
- j) Click .
- k) If using Legacy DVR software, also load ARLCipApps by double clicking ARLCipApps_20161130_1113.msi (see Fig. 3).
- l) The ARLCipApps Setup Wizard appears on the screen. It is similar to the ARLApps Wizard shown in Fig. 4.
- m) Follow steps 2e through 2k to install the applications.

3.3 Load the DSPro Software

The following are the steps for loading the DSPro software:

- 1) Copy Nomads.7z from the \\Synologynas\Root\Public\IOAIDE setup to the desktop:

- a) Right click the Windows icon  and select . The File Explorer popup window appears on the screen.
- b) In the File Explorer textbox at the top, type (including the spaces) **\\Synologynas\Root\Public\IOAIDE setup**
- c) Drag Nomads.7z to the desktop. Unzip the files to the desktop. Right click on Nomads.7z folder and click 7-Zip->Extract Here (Fig. 6).

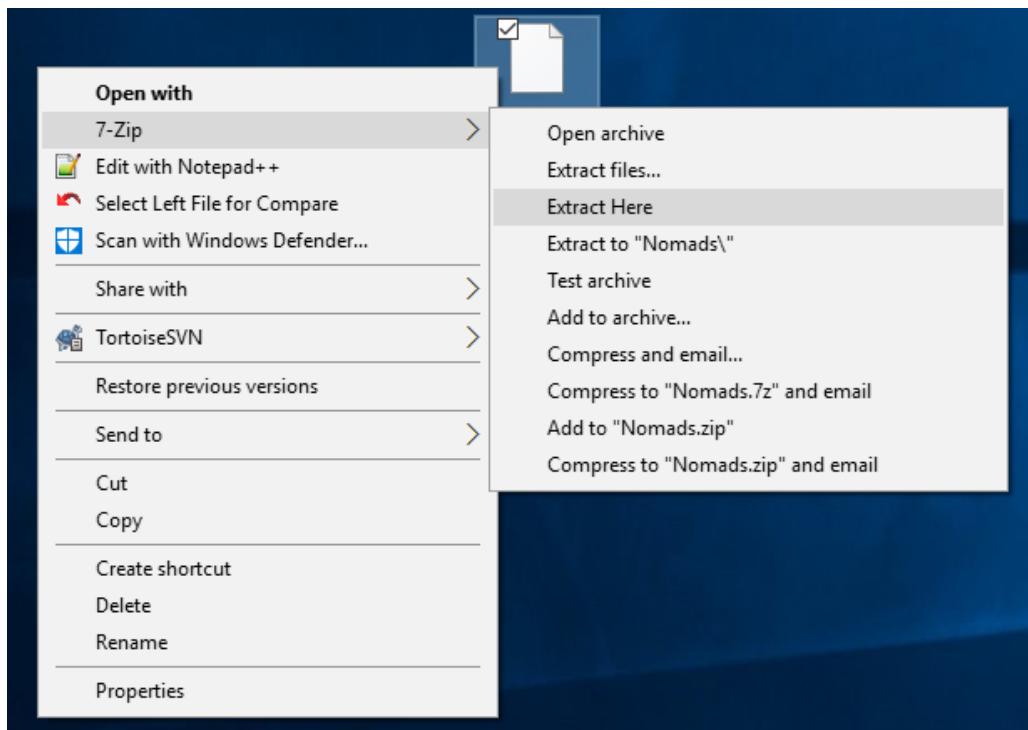

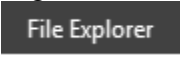


Fig. 6 Unzip files

- d) When the unzip process is complete, move the unzipped Nomads folder to C:

- i) Right click the Windows icon  and select . The File Explorer popup window appears on the screen (Fig. 7).

- ii) Drag the Nomads folder over **Local Disk (C:)** and release as shown in Fig. 7. The folder has now been moved to the C drive.

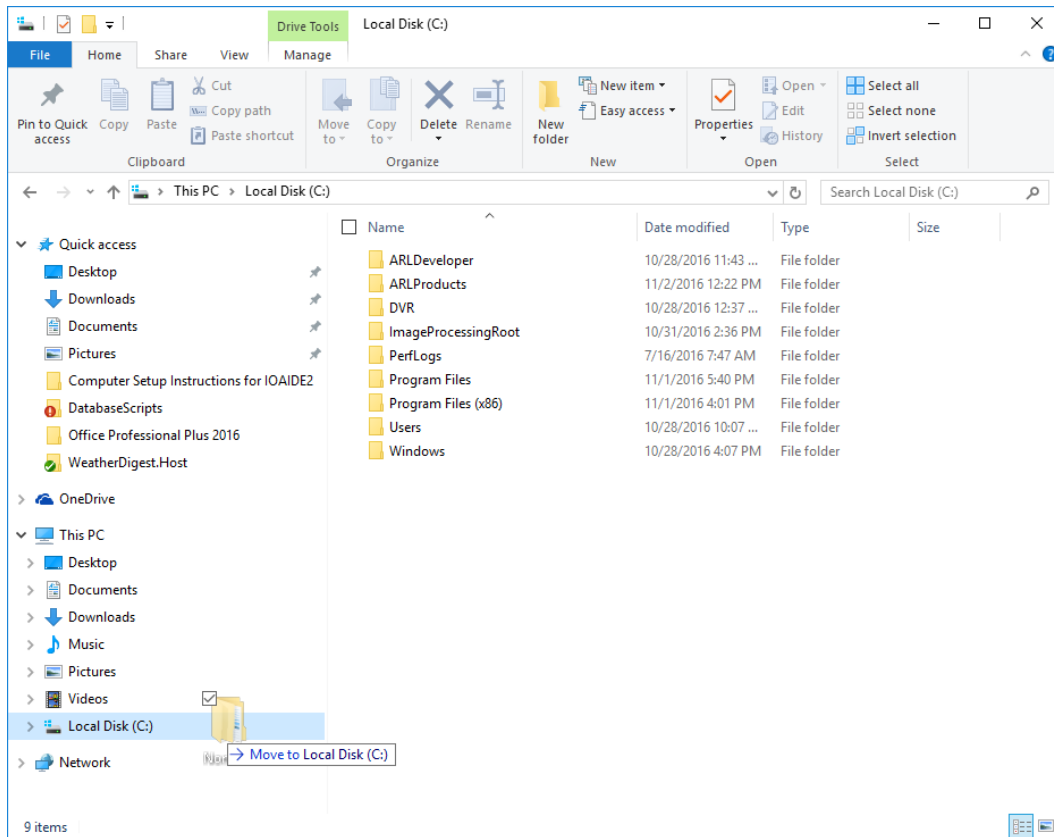



Fig. 7 Move Nomads to C:

- 2) Edit the DSPPro config file to reflect the computer name:

- Click the Windows icon  and select **File Explorer** on the right. The File Explorer popup window appears on the screen (see Fig. 7).
- Right click **This PC**, select Properties. The system popup window appears on the screen (Fig. 8).
- Find the computer name. In this example, the computer name is “juliette”.

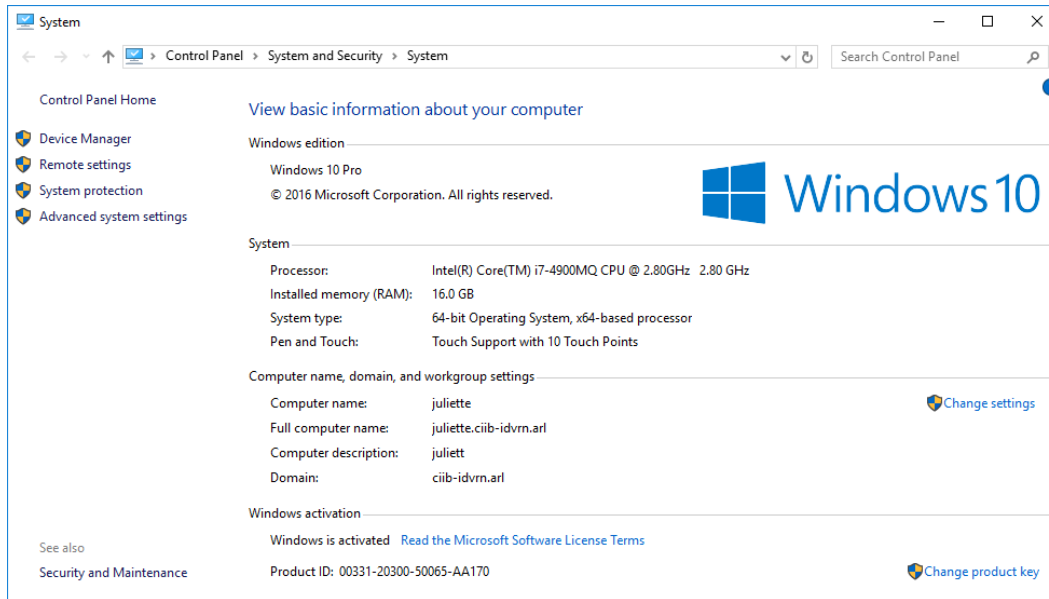





Fig. 8 System popup window

- d) Using the File Explorer as in step 2a, navigate to C:\ARLDeveloper\Components\ihmc\nomads\aci\conf by typing the previous string into the textbox at the top of the File Explorer.
- e) Open onrdemo-sen.properties with Notepad (double click onrdemo-sen.properties and it opens on the desktop). If this does not open, right select  or More apps in the How do you open this file? popup window. If selecting More apps, choose  from the list of choices.
- f) Change first line aci.disService.nodeUUID=juliette to aci.disService.nodeUUID='computer name' where “computer name” is the name found in step 2c (i.e., juliette). This computer name is also used in step 9h.
- g) Save file right click File and select Save. Then exit NotePad by clicking the  in the upper-right corner of the window.

3) Build the DSPro software package:

- a) Using the File Explorer as in step 2a, navigate to C:\Nomads\aci\cpp\dspro2\win32.

- b) Double click dsro2.sln or dspro2 with type defined as Microsoft Visual Studio Solution. This opens the DSPro solution in Microsoft Visual Studio (Fig. 9).

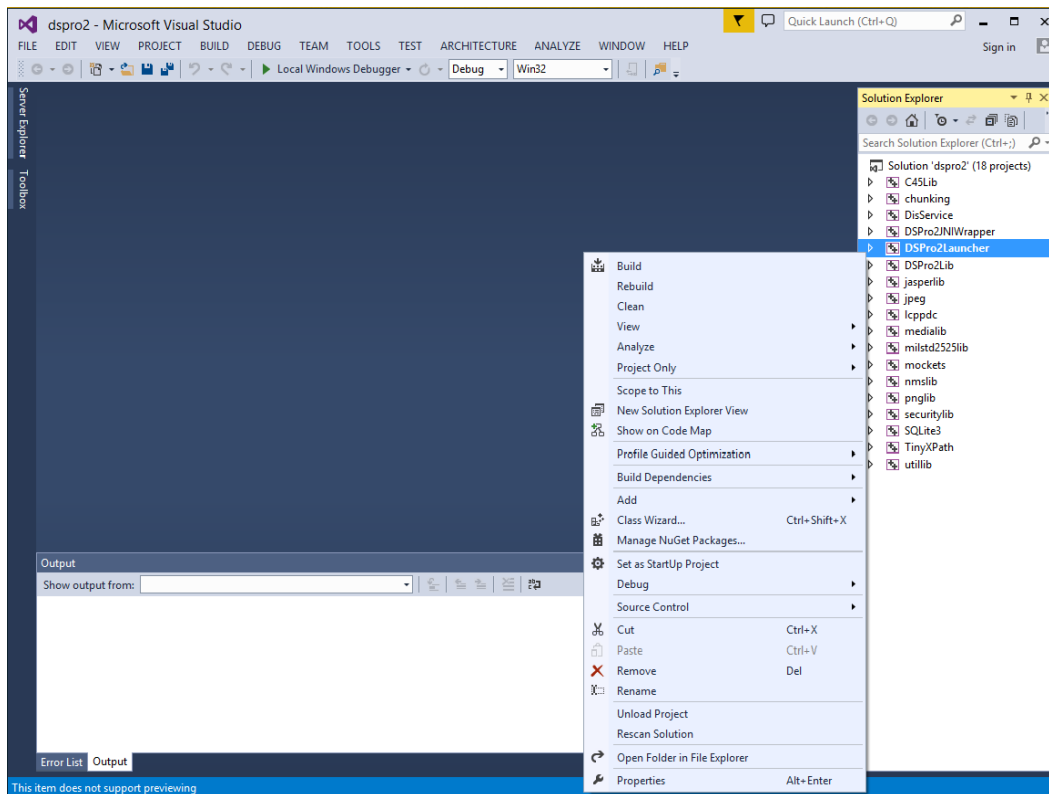




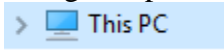







Fig. 9 Visual Studio DSPro project

- c) Make sure dsPro2Launcher is set as the startup program. If **dsPro2Launcher** is in bold letters in the Solution Explorer window, it is set as the startup program. If this is not the case, right click dsPro2Launcher and select “Set as Startup Project”.
- d) Right click dsPro2Launcher and select Build to build the project.
- e) The Output window should display Build: 18 succeeded, 0 failed, 0 up-to-date, 0 skipped.
- f) Exit Visual Studio by clicking the  in the upper-right corner.
- 4) Set up to use Bing Maps:
- a) Copy  **Bing Maps WPF Control** from the `\\Synologynas\root\public\IOAIDE setup` to `C:\ProgramFiles(x86)`.
- b) Using the File Explorer as in step 2a, navigate to `\\Synologynas\root\public\IOAIDE setup`.

- c) Using a separate File Explorer as in step 2a, navigate to C:\ProgramFiles(x86).
 - d) Drag  Bing Maps WPF Control from \\Synologynas\root\public\IOAIDE setup to C:\ProgramFiles(x86).
- 5) For Legacy video and stored tracker video, perform the following:
- a) Copy the  DVR from the \\Synologynas\root\public\IOAIDE setup to the C: drive:
 - i) Using the File Explorer as in step 2a, navigate to \\Synologynas\root\public\IOAIDE setup.
 - ii) Using a separate File Explorer as in step 2a, double click  .
 - iii) Drag the  DVR to  Local Disk (C:) 344 GB free of 465 GB .
 - b) If the shortcut for the DVR  is not on the desktop, execute steps c through e.
 - c) Using the File Explorer. Navigate to C:\DVR.
 - d) Right click on dvr.dvrconfig and click on Create shortcut.
 - e) Move the shortcut to the desktop.
- 6) For Salience Video data, do the following:
- a) Copy  Salience Video from \\Synologynas\root\public\IOAIDE setup to the desktop. See instructions for copying files in step 5.
- 7) Add Unified Database using SQLServer Management Studio:

- a) Click the Windows icon  ->Microsoft SQL Server 2016->SQL Server Management Studio. The Connect to Server window appears on the screen. For the purposes of this report, the server name “juliette” is used as an example server name where IOAIDE is being installed, and in this case, also where the SQL Server databases reside (Fig. 10).

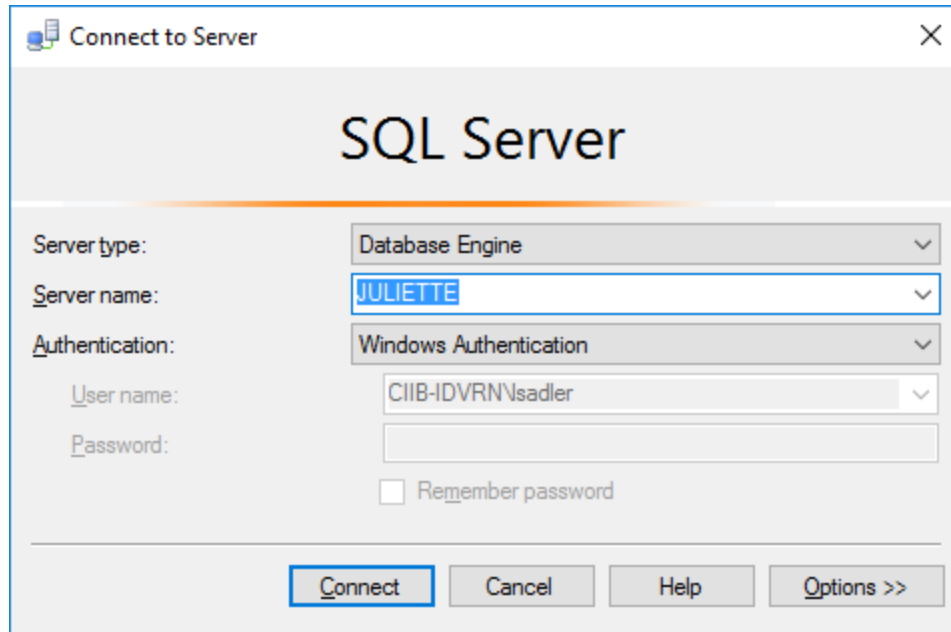


Fig. 10 Connect to Server window

b) Click .

c) In the upper-left corner of the SQL Server Management Studio window, click File->Open->File... (Fig. 11).

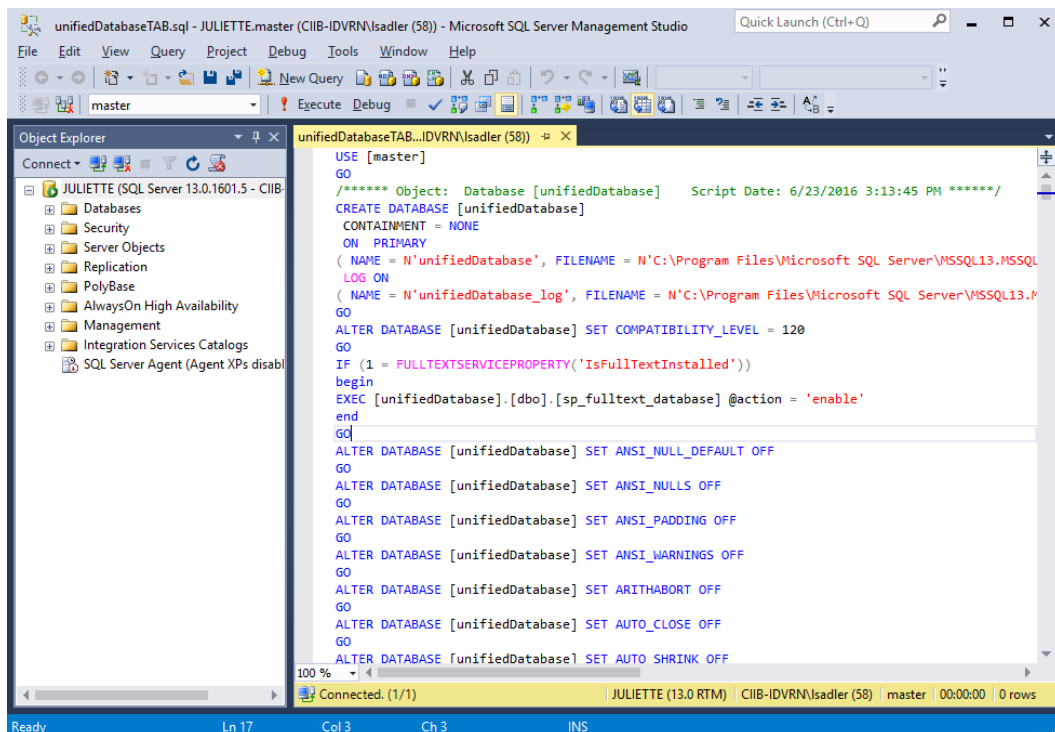
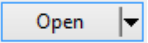


Fig. 11 SQL Server Script window

- d) Navigate to C:\ARLDeveloper\Components\DatabaseScripts.
- e) Click unifiedDatabase12-9-2016.sql. Click .
- f) The script appears in a window. Click in the script window.
- g) In the upper-left corner of the screen, navigate to Edit->Find and Replace->Quick Replace (see Fig. 11).
- h) The Find and Replace Window appears in the upper-right corner of the SQL Server Window (Fig. 12).

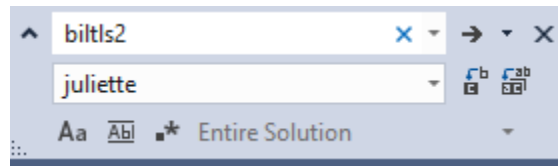

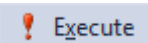


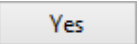


Fig. 12 Find and Replace window

- i) Type “bittls2”, in the find top textbox and the name of the target computer (juliette, in this example) in the replace textbox just below (see Fig. 12).
 - j) Select .
 - k) Click  to run the script. This builds the database and as well as inserts the demo data (see Fig. 11).
 - l) Remember the name of the server in the top corner. It is needed for subsequent steps. In this example, the server is juliette (see Fig. 11).
 - m) Exit SQL Server Management Studio by clicking the  in the upper-right corner (see Fig. 11).
- 8) Install ImageProcessing WebServices:
- a) Run MCRinstaller:
 - i) Navigate to C:\ARLDeveloper\COTS\MatLabCompiler Runtime_7.17 and double click  **MCRInstaller.exe**.
 - ii) The User Account Control window appears on the screen requesting permission to make changes to this computer (see Fig. 5).
 - iii) Click  in the User Account Control window.

- iv) The MATLAB Compiler Runtime Installer window appears on the screen (Fig. 13).

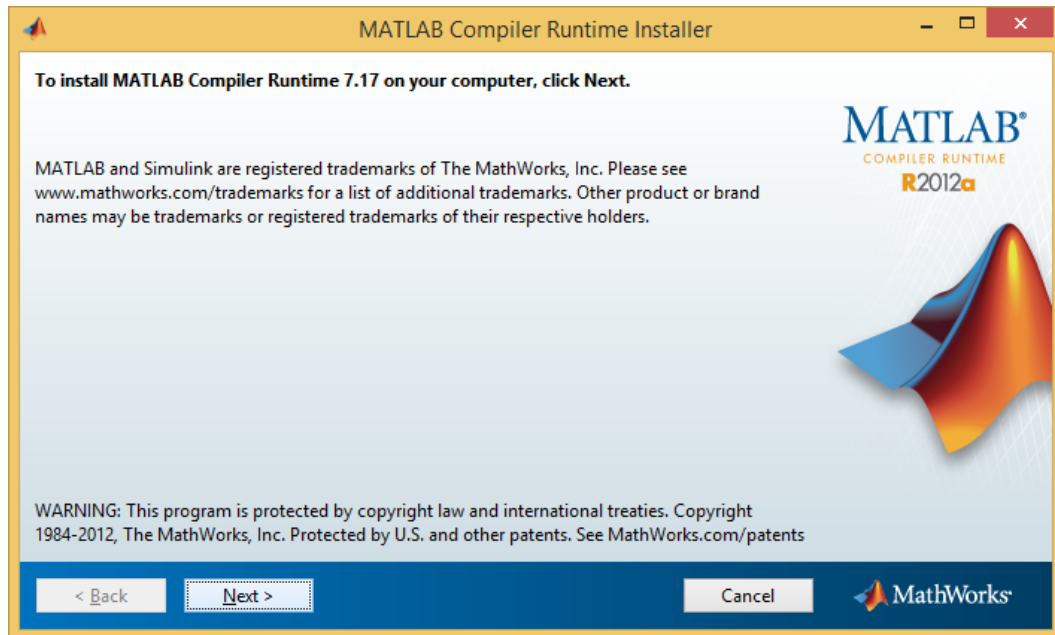
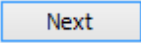
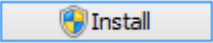


Fig. 13 MATLAB Compiler Runtime Installer

- v) Click  for all subsequent setup windows, using the defaults.
- vi) Click .
- b) Copy the folder: \\SynologyNAS\root\Public\cschlesiger\ ARLImage Processing\ImageProcessingRoot to C:\ImageProcessingRoot.
- c) In the Explorer window, navigate to C:\ARLDeveloper\Components\ ARLImageProcessing.Host. Double click ARLImageProcessing.sln to open the Visual Studio project (Fig. 14).

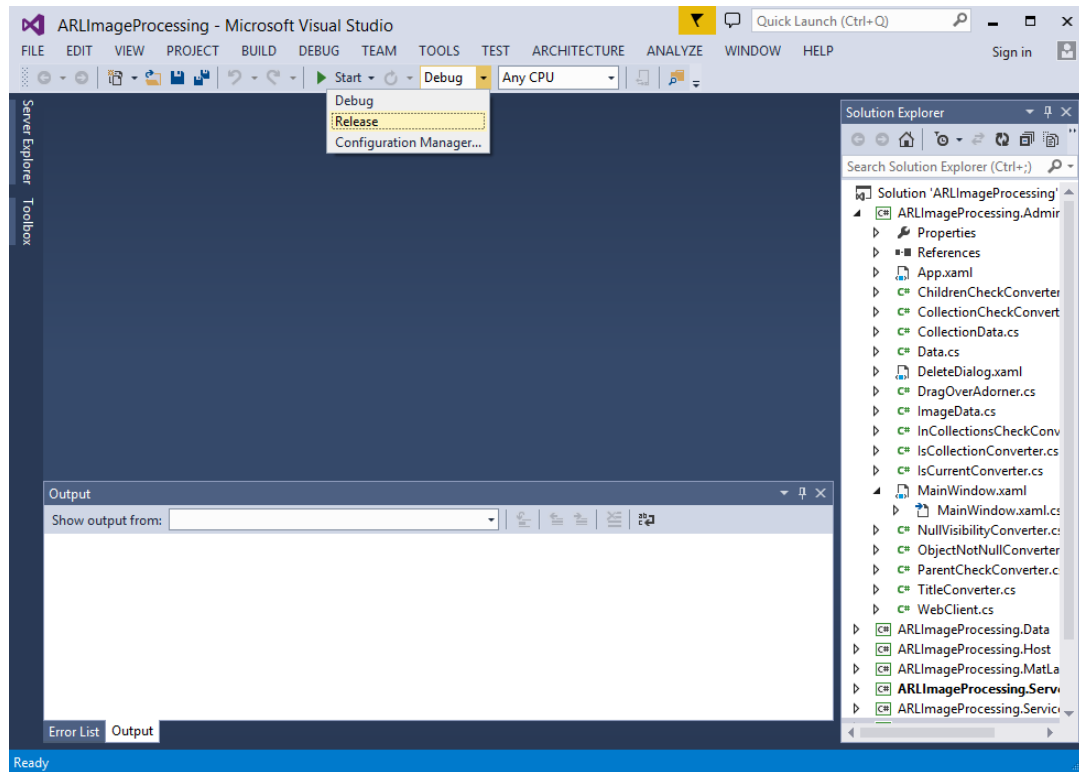

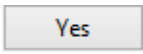


Fig. 14 Visual Studio: ARLImageProcessing.Host

- d) Set build parameters to Release and Any CPU. Build the entire solution using the BUILD->Build Solution button at the top. This builds the project in Release mode, which places it in a folder designated for use by the Windows Service. Be sure to return to Debug mode for subsequent steps involving this Visual Studio project (see Fig. 14).
- e) Open up the Developer Command Prompt for Visual Studio with Administrator privileges. To do this, click the Windows icon  ->Visual Studio 2015 folder and right click Developer Command Prompt for ... and left click ->More->Run as administrator.
- f) A User Account Control window appears on the screen (see Fig. 5). Click  in the User Account Control window. The Developer Command Prompt for VS2015 window appears on the screen (Fig. 15).

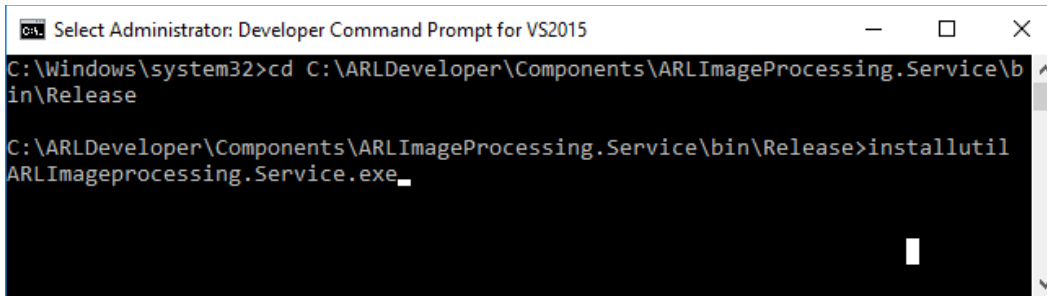





Fig. 15 Developer Command Prompt for VS2015

- g) In the Developer Command Prompt for VS2015 window, navigate to the folder by typing “cd C:\ARLDeveloper\Components\ARLImageProcessing.Service\bin\Release”, as shown in the first command line in Fig. 15.
- h) Run the command: installutil ARLImageprocessing.Service.exe, as shown in the second command line in Fig. 15. The command should indicate success. Click the  in the upper-right corner to exit the command prompt.
- i) Start the web service by going to Services (right click the Windows icon  ->Control Panel->System and Security. In the popup window, select Administrative Tools. In the next popup window, select Services.) In the Services window, look for ARL Image Processing Web Service. It should already be set to start Automatic on reboot. To start it now, after the initial install, select it then click the Play button  at the top of the window (or right click and select Start from the context menu) (Fig. 16).

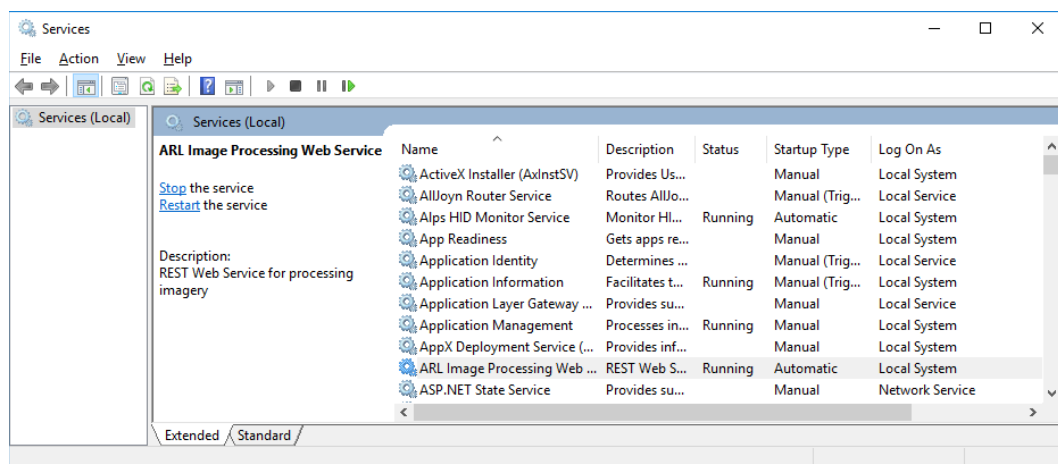



Fig. 16 Services

- j) Verify the service is running by launching the Admin tool:
- I. Using the File Explorer, navigate to C:\ARLDeveloper\Components\ARLImageProcessing.Admin\bin\Release.
 - II. Double click ARLImageProcessing.Admin Application to execute. The ARL Image Processing Web Server Administration window appears on the screen. At the top of the ARL Image Processing Web Server Administration window, the address in the Current Server textbox should be http://localhost:8080/ARLImageProcessing (Fig. 17).
 - III. Click  at the top right of the window. If the window populates with images and collections as shown on the left side of the window, the install was successful (Fig. 17).

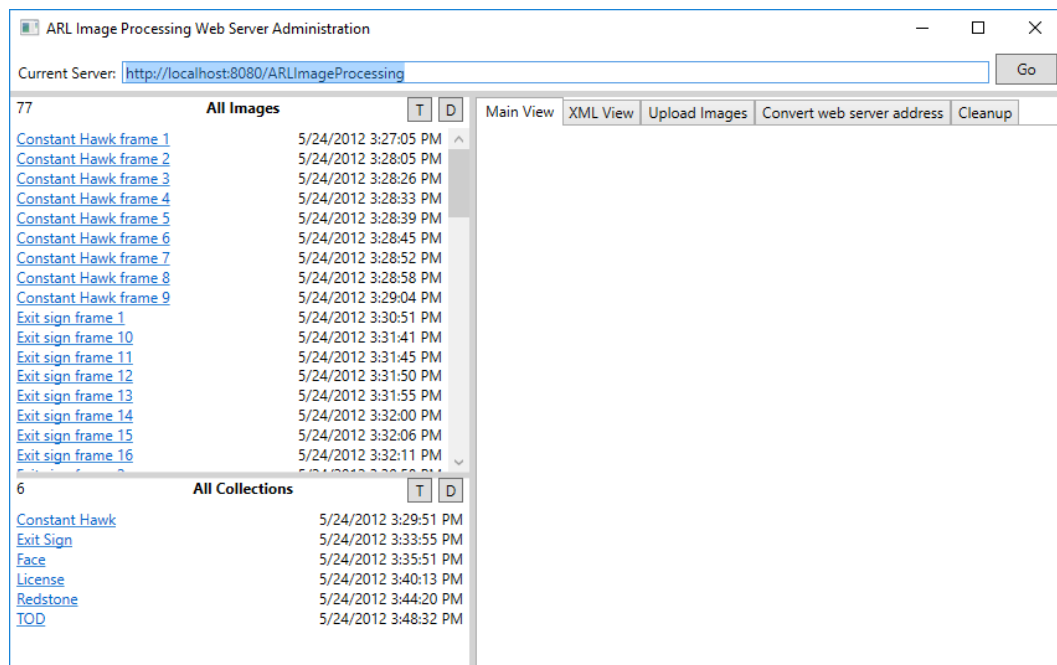


Fig. 17 ARL Image Processing Web Server Administration

- 9) Install the Weather Digest WebServices:
- a) Install ASP.NET MVC4:
 - i) Navigate to C:\ARLDeveloper\COTS\AspNetMVC4Setup.exe and double click  AspNetMVC4Setup.exe .

- ii) On the first window that appears, click on the checkbox to agree to the license terms and conditions (Fig. 18).
- iii) Click on Install.

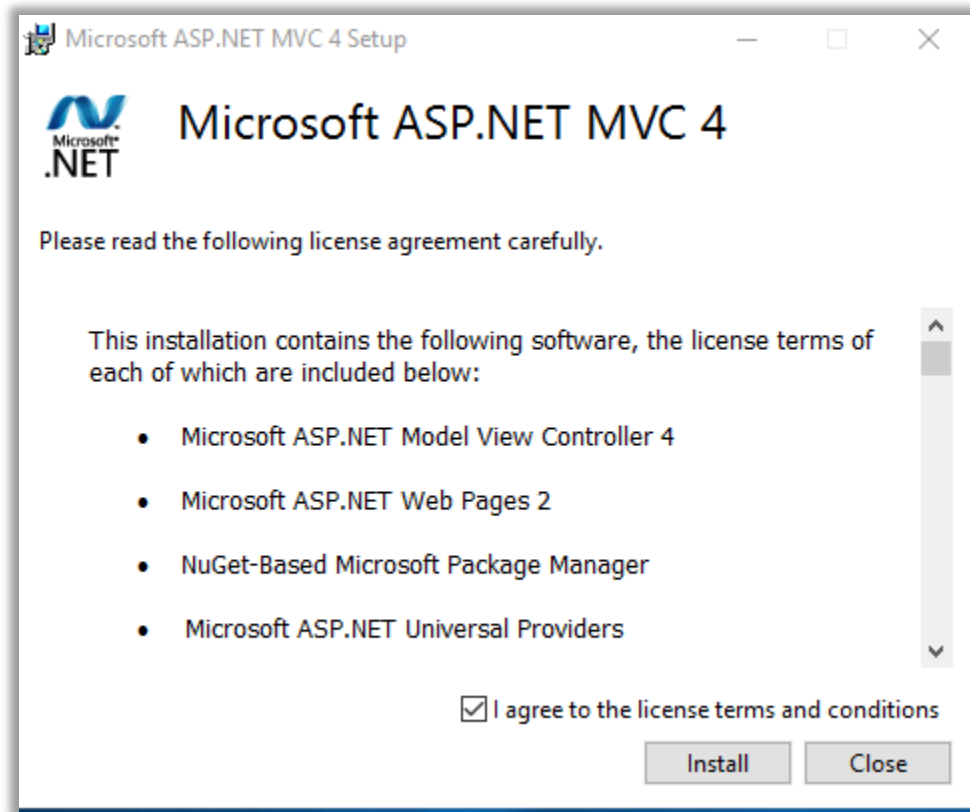
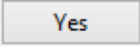


Fig. 18 Microsoft ASP.NET MVC 4 installation

- iv) The User Account Control window appears on the screen requesting permission to make changes to this computer (see Fig. 5).
 - v) Click  in the User Account Control window.
 - vi) Close the installer when it is complete.
- b) Ensure the Tweetinvi COTS library has been checked out by verifying that the following folder and file exist:
C:\ARLDeveloper\COTS\Tweetinvi\1.0\output\Tweetinvi.dll.

- c) In the Explorer window, navigate to C:\ARLDeveloper\Components\WeatherDigest.Host. Double click WeatherDigest Solution to open the visual studio project (Fig. 19).

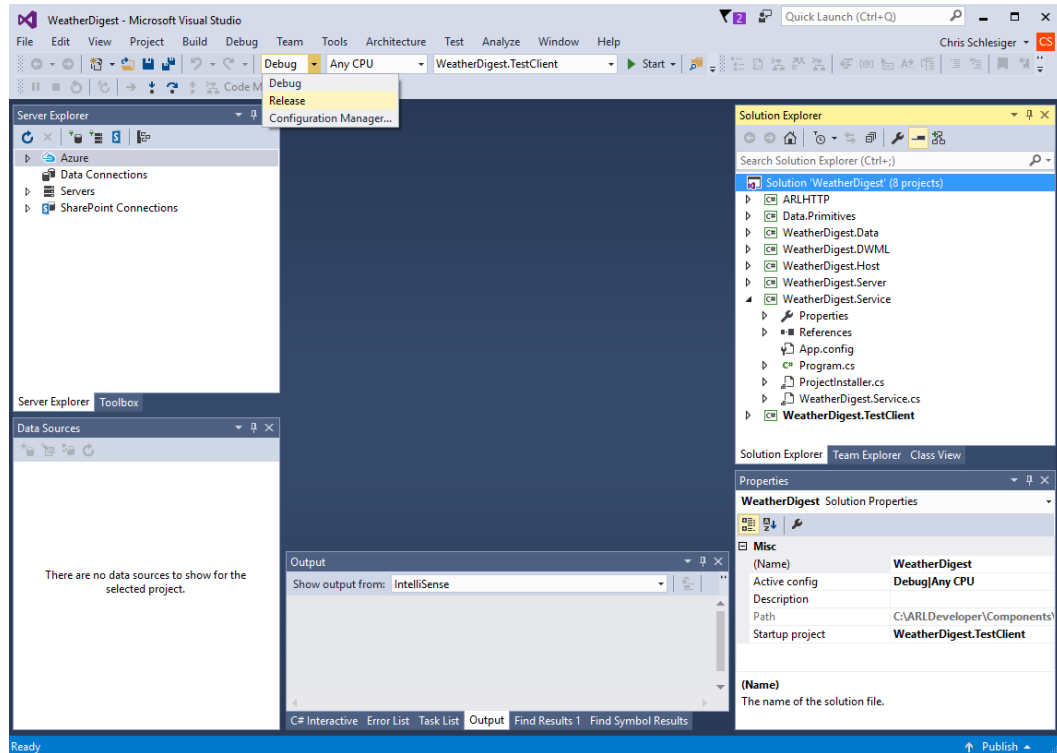

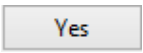

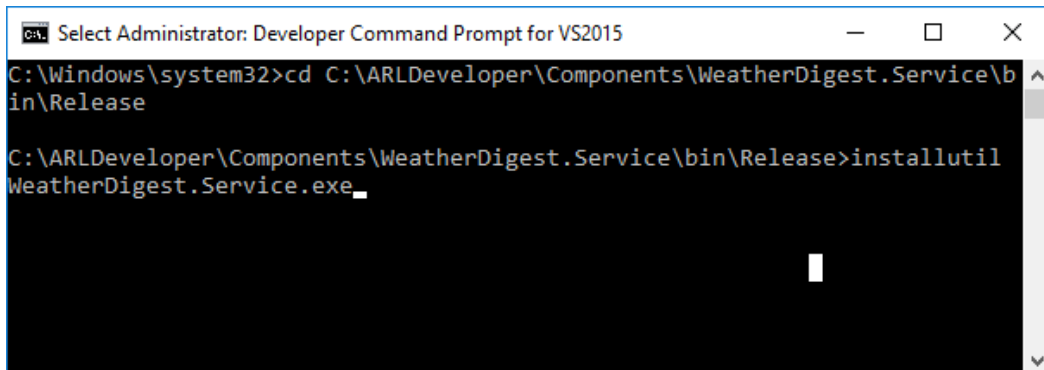


Fig. 19 Visual Studio: WeatherDigest.sln

- d) First, leave the build parameters set to Debug and Any CPU. Build the entire solution using the BUILD->Build Solution button at the top. This builds the project in debug mode, copying some important files to the bin\Debug folder.
- e) Next, set the build parameters to Release and Any CPU. Build the entire solution using the BUILD->Build Solution button at the top. This builds the project in Release mode, which places it in a folder designated for use by the Windows Service. Be sure to return to Debug mode for subsequent steps involving this Visual Studio project (see Fig. 19).
- f) In the File Explorer window, navigate to C:\ARLDeveloper\COTS\Tweetinvi\1.0. Copy the following 4 files and paste them into the C:\ARLDeveloper\Components\WeatherDigest.Service\bin\Release folder:

- i) Autofac.dll

- ii) Microsoft.Threading.Tasks.dll
 - iii) Microsoft.Threading.Tasks.Extensions.dll
 - iv) Newtonsoft.Json.dll
- g) Open up the Developer Command Prompt for Visual Studio with Administrator privileges. To do this, click the Windows icon  -> Visual Studio 2015 folder and right click Developer Command Prompt for ... and left click -> More->Run as administrator.
- h) A User Account Control window appears on the screen (see Fig. 5). Click  in the User Account Control window. The Developer Command Prompt for VS2015 window appears on the screen.
- i) In the Developer Command Prompt for VS 2015 window navigate to the folder by typing “cd C:\ARLDeveloper\Components\WeatherDigest.Service\bin\Release”, as shown in the first command line in Fig. 20.
- j) Run the command: installutil WeatherDigest.Service.exe, as shown in the second command line in Fig. 20. The command should indicate success. Click the  in the upper-right corner to exit the command prompt.




```


C:\Windows\system32>cd C:\ARLDeveloper\Components\WeatherDigest.Service\bin\Release

C:\ARLDeveloper\Components\WeatherDigest.Service\bin\Release>installutil WeatherDigest.Service.exe

```

Fig. 20 Developer Command Prompt for VS2015

- k) Start the web service by going to Services (Right click the Windows icon  -> Control Panel->System and Security. In the popup window, select Administrative Tools. In the next popup window, select Services). In the Services window, look for ARL Weather Digest Web Service. It should already be set to start Automatic on

reboot. To start it now, after the initial install, select it then click the Play button  at the top of the window (or right click and select Start from the context menu) (Fig. 21).

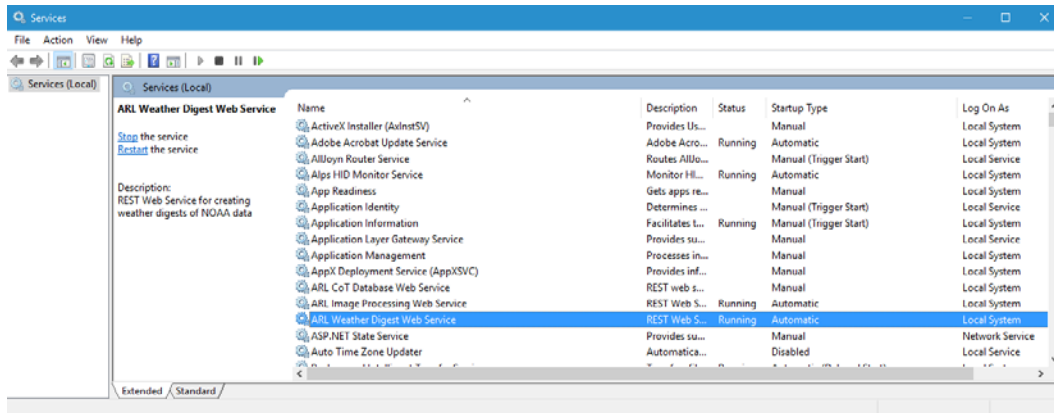


Fig. 21 Services

1) Verify the service is running by launching a web browser (e.g., Internet Explorer):

i) In the address bar of a new tab, type

<http://localhost:8040/WeatherDigest/Digest?lat=38.9894&lon=-76.9729>

ii) The resulting output should resemble the output shown in Fig. 22.

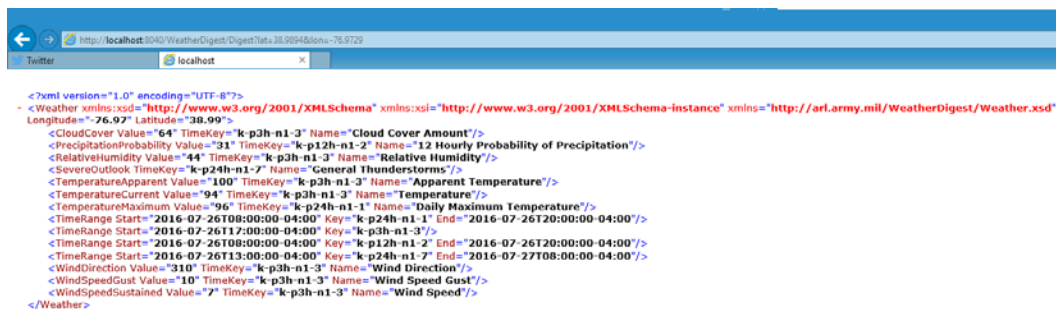


Fig. 22 Internet Explorer Weather Digest test

iii) Verify that tweets were sent out by logging into Twitter at <https://twitter.com>. The login name is ARLWeatherDigest@gmail.com and the password is provided by the administrator. There should be tweets corresponding to the date and time of the test just run similar to what is shown in Fig. 23.

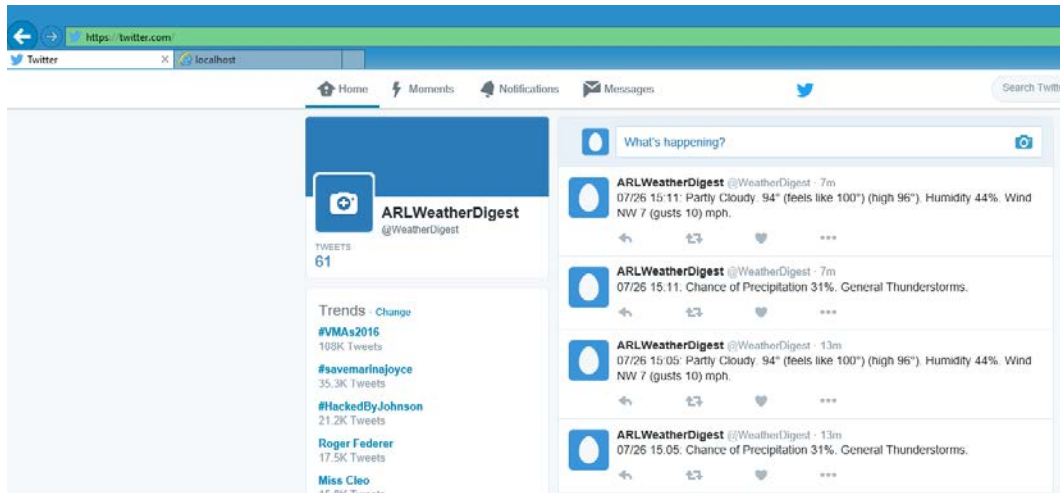

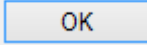


Fig. 23 Internet Explorer Weather Digest Twitter test

3.4 Update Java Setting in Regedit

The following are the steps for updating the Java setting so IOAIDE can navigate the web:

- 1) Right click the Windows icon , select Run, type **regedit** in the Open: textbox, and then click . The User Account Control popup Window appears on the screen (Fig. 24).

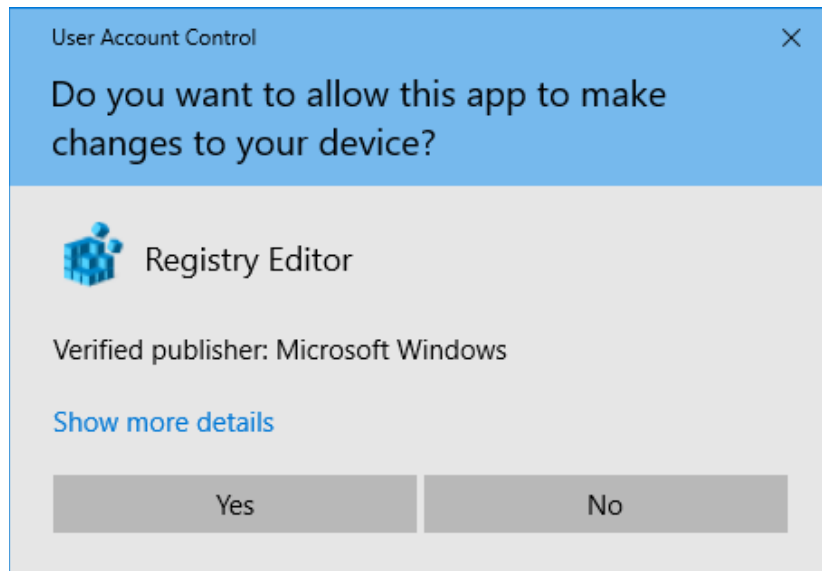


Fig. 24 User Account Control for regedit

- 2) Click **Yes**. The Registry Editor popup window appears on the screen (Fig. 25).

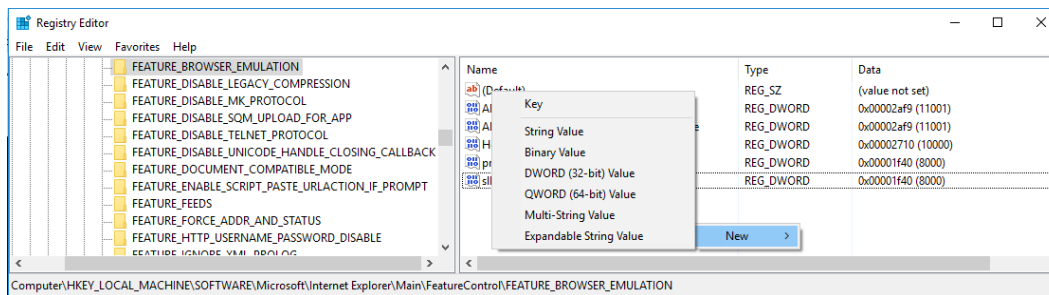


Fig. 25 Registry Editor

- 3) In the left-most panel, navigate to HKEY_LOCAL_MACHINE->SOFTWARE->Microsoft->Internet Explorer->Main->FeatureControl->FEATURE_BROWSER_EMULATION (see Fig. 25).
- 4) In right panel, right click and select New->DWORD (32-bit) Value (see Fig. 25).
- 5) Modify **New Value #1** to ARLImageProcessingSurfaceApp.exe.
- 6) Right click ARLImageProcessingSurfaceApp.exe and click Modify....
- 7) The Edit DWORD (32-bit) Value window appears on the screen (Fig. 26).
- 8) Type **2af9** in Value data textbox (Fig. 26).
- 9) Click **OK**.

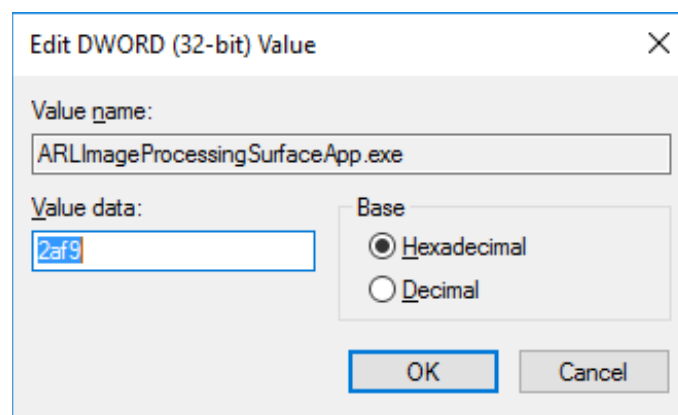

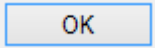



Fig. 26 Edit DWORD (32-bit) Value

- 10) In right panel, right click and select New->DWORD (32-bit) Value again (see Fig. 25).

- 11) Modify  to ARLImageProcessingSurfaceApp.vshost.exe.
- 12) Right click ARLImageProcessingSurfaceApp.exe and click “Modify....”
- 13) The Edit DWORD (32-bit) Value window appears on the computer screen.
Type **2af9** in Value data textbox (see Fig. 26).
- 14) Click  (see Fig. 26).
- 15) Exit the Registry Editor by clicking the  in the upper-right corner.

3.5 Build the IOAIDE Software Packages

The following are the steps for building the IOAIDE software packages:

- 1) Using the File Explorer, navigate to C:\ARLDeveloper\Components\ARLImageProcessing.Win8App. Double click ARLImageProcessing.SurfaceApp.sln or ARLImageProcessing.SurfaceApp with the type defined as Microsoft Visual Studio Solution to open the project in Visual Studio (Fig. 27).
- 2) In the Solution Explorer window on the right, make sure ARLImageProcessing.SurfaceApp is set as the startup project. **ARLImageProcessing.SurfaceApp** is in **bold** letters if it is set as the Startup project. If not, then right click ARLImageProcessing.SurfaceApp and select “Set as Startup Project” (Fig. 27).
- 3) Set the build configuration Debug and x64 as shown in Fig. 27.
- 4) Right click ARLImageProcessing.SurfaceApp and select Build (Fig. 27).

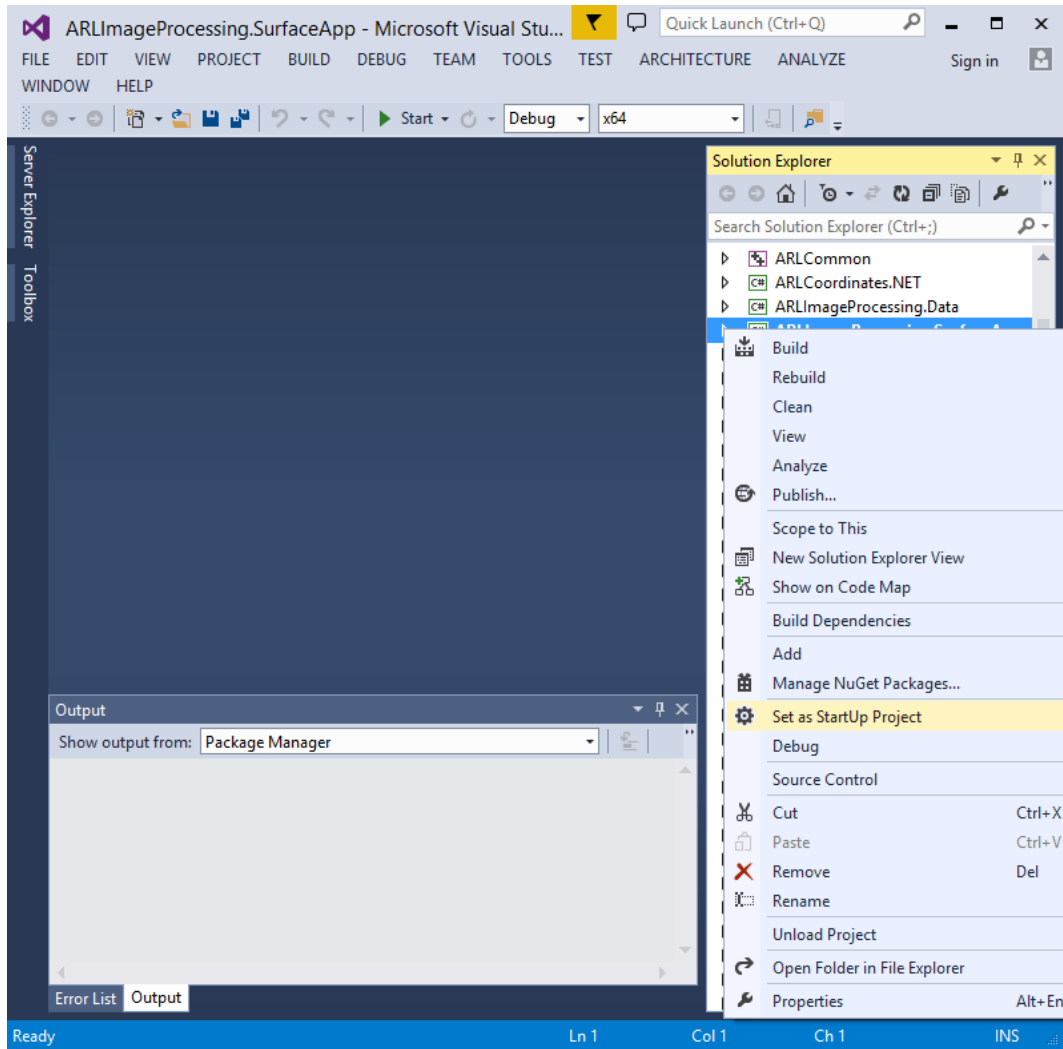




Fig. 27 Visual Studio: ARLImageProcessing.SurfaceApp

- 5) Exit Visual Studio by clicking the  in the upper-right corner.

3.6 Build DSProTest


The following are the steps for building DSProTest:

- 1) Using the File Explorer, navigate to C:\ARLDeveloper\Components\DSProTest. Double click the DSProTest.sln or DSProTest with the type defined as Microsoft Visual Studio Solution to open the project in Visual Studio.
- 2) In the Solution Explorer window on the right, make sure DSProTest is set as the startup project. **DSProTest** is in **bold** letters if it is set as the Startup project. If not, right click DSProTest and select 'Set as Startup Project'.

- 3) Set the build configuration Debug and x64 as shown in Fig. 27.
- 4) Right click **DSProTest** and select Build. Fig. 27.
- 5) Exit Visual Studio by clicking the  in the upper-right corner.

3.7 Copy IOAIDE and Dependencies Shortcuts to Executables

The following are the steps for copying the IOAIDE and dependencies shortcuts to create executables:


- 1) Copy  **IOAIDE Startup** from \\Synologynas\root\public\IOAIDE setup to the desktop.
- 2) See the instructions for copying files in Section 3.3, step 5.

4. Execute IOAIDE and Dependencies in Development/Debug Mode

This section describes each step necessary to execute IOAIDE and dependencies software in a development environment. A demonstration can also be given in this mode of operation.

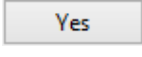
- 1) Start the DVR:
 - a) If using stored data (needed for the anomaly detection demo) for the video (not necessary for .avi files.), on the desktop, double click



to start the network and double click  to start the dvr client.

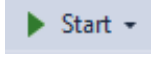
- 2) Run DSPro:
 - a) Delete the DSPro database from previous runs if it exists. Using the File Explorer, navigate to C:\Nomads\aci\cpp\dspro2\win32. Delete the file with the extension .sqlite.
 - b) In this directory, double click dsro2.sln to open the DSPro project in Visual Studio (see Fig. 9). Make sure dsPro2Launcher is the startup project. **dsPro2Launcher** should be in **bold** if it is the StartUp Project. If it is not, right click dsPro2Launcher and select Set as StartUp Project. Run dsPro2Launcher in the debugger by clicking

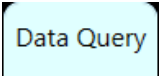

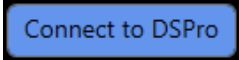


c) Click , when asked to build projects that are out of date.

3) Run IOAIDE:

a) Navigate to C:\ARLDeveloper\Components\ARLImageProcessing.Win8App. Double click ARLImageProcessing.SurfaceApp.sln to open it in Visual Studio (see Fig. 27).

b) Run IOAIDE in the debugger by clicking .


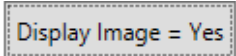
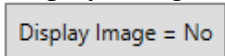
c) In the Data Query tab , open a DSPro Window by clicking  and connect to DSPro by clicking .

4) Run DSProTest:

a) If it is desired that new incoming data from sensors or Android devices be stored in the Unified database to be queried by IOAIDE, then DSProTest must also be running.

b) Navigate to C:\ARLDeveloper\Components\DSProTest. Double click DSProTest.sln to open it in Visual Studio (see Fig. 27).

c) Run DSProTest in the debugger by clicking .

d) In the MainWindow of DSProTest, connect to DSPro by clicking  and set Display Image = No by clicking . It now says .

5. Execute IOAIDE and Dependencies from Shortcuts and Executables

If the IOAIDE startup folder does not exist on the desktop, it is necessary to make the shortcuts before starting:

1) Create a new folder on the desktop entitled “IOAIDE Startup”.

2) Make a shortcut for IOAIDE:

a) In the File Explorer window in Section, 3.3, step 2a, navigate to C:\ARLDeveloper\Components\ARLImageProcessing.Win8App\bin\x64\Debug.

- b) Right click ARLImageProcessing.SurfaceApp.exe and select Create shortcut.
- c) Copy ARLImageProcessing.SurfaceApp.exe-Shortcut to the IOAIDE Startup folder on the desktop.

3) Make a shortcut for DSPro2Launcher:

- a) In the File Explorer window in Section 3.3, step 2a, navigate to C:\Nomads\aci\cpp\dspro2\win32\Debug.
- b) Right click on the Application DSPro2Launcher and select Create shortcut.
- c) Right click DSPro2Launcher-Shortcut and select Properties.
- d) The Target: Textbox should contain: C:\Nomads\aci\cpp\dspro2\win32\Debug\DSPro2Launcher.exe.
- e) Add the following text:

```
-c C:\Nomads\aci\conf\onrdemo-sen.properties
-e C:\Nomads\aci\conf\dspro.metadata.extraattributes.xml
-v C:\Nomads\aci\conf\dspro.metadata.extravalues.xml
```

- f) The Target: Textbox should now contain the following:

```
C:\Nomads\aci\cpp\dspro2\win32\Debug\DSPro2Launcher.exe
-c C:\Nomads\aci\conf\onrdemo-sen.properties
-e C:\Nomads\aci\conf\dspro.metadata.extraattributes.xml
-v C:\Nomads\aci\conf\dspro.metadata.extravalues.xml
```

- g) Click .

- h) Copy the DSPro2Launcher-Shortcut to the IOAIDE Startup folder on the desktop.

4) Make a shortcut for DSProTest:

- a) In the File Explorer window in Section 3.3, step 2a, navigate to C:\ARLDeveloper\Components\DSProTest\bin\x64\Debug.
- b) Right click DSProTest.exe and select Create shortcut.
- c) Copy DSProTest.exe-Shortcut to the IOAIDE Startup folder on the desktop.

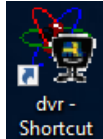
This following describes each step necessary to execute IOAIDE and dependencies software in a user/demonstration/experiment mode/environment:

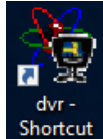
1) Start the DVR:

- a) If using stored data (needed for the anomaly detection demo) for the video (not necessary for .avi files.), on the desktop, double click



to start the network.



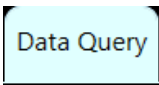

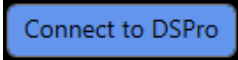
- b) Double click  to start the dvr client.

2) Run DSPro:

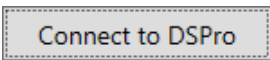
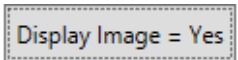
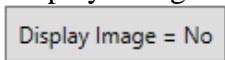
- a) Open the IOAIDE Startup folder on the desktop.
- b) Click the DSPro2Launcher shortcut icon.

3) Run IOAIDE:

- a) Open the IOAIDE Startup folder on the desktop.
- b) Click the IOAIDE shortcut icon.

- c) In the Data Query tab , open a DSPro window by clicking  and connect to DSPro by clicking .

4) Run DSProTest:

- a) If it is desired that new incoming data from sensors or Android devices be stored in the Unified database to be queried by IOAIDE, then DSProTest must also be running.
- b) Open the IOAIDE Startup folder on the desktop.
- c) Click the DSProTest shortcut icon.
- d) In the MainWindow of DSProTest, connect to DSPro by clicking  and set Display Image = No by clicking . It now says .

6. Conclusion

The IOAIDE application is still under development. It is designed to be extensible, allowing for the addition of new enhancement applications and algorithms to the existing modules as they become available. It can accommodate new research areas by adding additional modules. This report described all the steps required to initialize a computer with all the required software to develop and run the IOAIDE application in its current state. The installation process involves many steps due to dependencies on other software applications. This document will need to be updated with the addition of new software dependencies.

7. References

1. Sadler LC, Michaelis JR. An information management framework for C2 data analysis in tactical operations centers. Proceedings of the 21st International Command and Control Research and Technology Symposium; 2016 Sep 6–8; London, UK.
2. Winkler RP, Schlesiger C. Image processing REST web services. Adelphi (MD): Army Research Laboratory (US); 2013 Mar. Report No.: ARL-TR-6393.
3. Mazzini A, Stefanelli C, Tortonesi M, Benincasa G, Suri N. Disservice: network state monitoring and prediction for opportunistic information dissemination in tactical networks. In Military Communications Conference, 2010-MILCOM: IEEE; 2010. p. 555–560.
4. Suri N, Benincasa G, Lenzi R, Tortonesi M, Stefanelli C, Sadler L. Exploring value-of-information-based approaches to support effective communications in tactical networks. Communications Magazine. 2015;IEEE 53(10):39–45.

1 DEFENSE TECHNICAL
(PDF) INFORMATION CTR
DTIC OCA

2 DIRECTOR
(PDF) US ARMY RESEARCH LAB
RDRL CIO L
IMAL HRA MAIL & RECORDS
MGMT

1 GOVT PRINTG OFC
(PDF) A MALHOTRA

2 DIRECTOR
(PDF) US ARMY RESEARCH LAB
RDRL CII B
LC SADLER
CD SCHLESIGER